IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re patent application of:

Nimrod Megiddo — Atty. Docket No.: ARC920030085US1

Serial No.: 10/723,850 — Group Art Unit: 2191

Filed: November 25, 2003 — Examiner: Wu, Junchun

For: SYSTEM AND METHOD FOR AUTONOMIC OPTIMIZATION BY COMPUTER PROGRAMS

Honorable Commissioner of Patents
Alexandria, Virginia 22313-1450

APPELLANT'S BRIEF ON APPEAL UNDER 35 U.S.C. §134(a)

Sir:

Appellant respectfully appeals the decision of the Examiner in the final rejection of claims 1, 5-8, 11-13, 17-19, and 23-24 in the Final Office Action mailed March 6, 2009.

I. STATEMENT OF THE REAL PARTY OF INTEREST

The real party of interest is International Business Machines Corporation, assignee of 100% interest of the above-referenced patent application.

II. STATEMENT OF RELATED CASES

There are no other appeals or interferences known to Appellant, Appellant's legal representative or Assignee, which would directly affect or be directly affected by or have a bearing on the Board's decision on this appeal.

III. JURISDICTIONAL STATEMENT

The Board has jurisdiction under 35 U.S.C. 134(a). The Examiner mailed a final rejection on March 6, 2009, setting a three-month shortened statutory period for response. The time for responding to the final rejection expired on June 8, 2009 (June 6, 2009 fell on a Saturday). Rule 134. A notice of appeal was filed on June 8, 2009. The time for filing an appeal brief is two months after the filing of a notice of appeal. Bd.R. 41.37(c). The time for filing an appeal brief expires on August 8, 2009. The appeal brief is being filed on August 6, 2009.

IV. TABLE OF CONTENTS

V. TABLE OF AUTHORITIES

(Not applicable.)

VI. STATUS OF AMENDMENTS

An amendment filed April 30, 2009, was entered by the examiner.

VII. GROUNDS OF REJECTION TO BE REVIEWED

Rejection of claims 1, 5-8, 11-13, and 17-18 as being unpatentable under 35 U.S.C. 103(a) over Polak (U.S. Patent No. 6,226,627), in view of Li (U.S. Patent No. 6,625,550).

Rejection of claims 19 and 23-24 as being unpatentable under 35 U.S.C. 103(a) over Polak, in view of Li, and in further view of Christensen, et al. (U.S. Patent No. 5,333,304), hereinafter referred to as Christensen.

VIII. STATEMENT OF FACTS

1) The 3/6/2009 Communication states that "[i]nputting a selection command that selects one function from a list of pre- selected functions for input into said computer program at a point of choice determined by a programmer (Co1.7lines 54-59 *"If multiple actions are enabled, the dependency action system uses a selection rule to determine which of the enabled actions is to be executed. It is possible to use a selection rule based on random choice. A policy can be pre-established that prescribes which action of a set of enabled actions is to be selected for execution.* " & col. 12 line 66 "*The enabled action selection rule determines which of the enabled actions 620 and* 622 *is tried first. Similarly, if the action 620 does not update the storage location* 632, *but updates the storage location 630, the selection rule determines whether the action 640 or the action* 622 *is used for recovery. This behavior can be change by the designer of the dependency action system by adding additional dependencies.* "). (3/6/2009 Communication, p. 2, ll. 17-p. 3, ll. 7)

2) The 3/6/2009 Communication states that "• allowing a learning protocol comprised of learning instructions in said computer program (col. 4 lines 45-50 *"The method and system of this invention are derived from biological systems and*

organisms that achieve resilience and adaptability through redundant functional components, planning the use of these components to achieve a given task, and learning appropriateness of components from experience.") to track and reward said one function that is selected and to determine an approximate optimal policy of choice of operation of said computer program based on said selection command (col.8 lines 49-57 *"An enabled action selection rule is used to select one of several enabled actions for execution. Different selection rules can be used, including a trivial selection rule that picks one of the enabled actions at random. This invention specifies an enabled action preference policy that is described below with respect to FIG.* 13. ")." (3/6/2009 Communication, p. 3, ll. 8-18)

3) The 3/6/2009 Communication states that "[b]ut Polak does not disclose • each function from said list of pre- selected functions is associated with a reward." (3/6/2009 Communication, p. 3, ll. 19-21)

4) The 3/6/2009 Communication states that "[h]owever, Li explicitly discloses • wherein each function from said list of pre- selected functions is associated with a reward (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple*

reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle. ")." (3/6/2009 Communication, p. 3, ll. 20 – p. 4. ll. 4)

5) The 3/6/2009 Communication states that "[p]er claims 5, 17 the rejection of claims 1 and 13 are incorporated respectively and Li further discloses • comprising inputting a rule command that establishes a rule for said computer program on how to determine said approximate optimal choice of operation (col. 16 lines 55-57 *"self-optimizing machine readily and automatically generates these and other similar rules in computer-coded form ready for use as instant machine expert systems* ... ")." (3/6/2009 Communication, p. 4, ll. 10-15)

6) The 3/6/2009 Communication states that "[p]er claims 6, 18 the rejection of claims 1 and 13 are incorporated and Li further discloses • comprising inputting a reward command that provides a reward, at a point of choice, determined by a programmer, in said computer program, for said one function selected by said selection command, which results in said approximate optimal choice for optimizing said computer program (col. 16 lines 5-9 *"the computer can also*

provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle. ")." (3/6/2009 Communication, p. 4, ll. 16 – p. 5. ll. 4)

7) The 3/6/2009 Communication states that "[p]er claim 7 Polak discloses A method of optimizing a computer program, said method comprising: • specifying at least one point of choice, determined by a programmer, in said computer program (col.8 lines 11-15 *"This invention preferably uses a specific type of selection rule, called a preference policy, that is based on a preference relation that specifies if one action is to be preferred over another action")."* (3/6/2009 Communication, p. 5, ll. 5-11)

8) The 3/6/2009 Communication states that "[d]efining a set of alternate choices at each point of choice ([Abstract] *"if one action fails to produce a value to a storage location, other alternative actions may still be enabled and can be executed.* "), wherein said set of alternate choices include operational choices comprises:• inputting a selection command that selects one function from a list of pre-selected functions into said computer program (col.7 lines 54-57 *"If multiple*

actions are enabled, the dependency action system uses a selection rule to determine which of the enabled actions is to be executed. It is possible to use a selection rule based on random choice.")" (3/6/2009 Communication, p. 5 ll. 12-19)

9) The 3/6/2009 Communication states that "[a]llowing a learning protocol comprised of learning instructions in said computer program (col. 4 lines 45-50 *"The method and system of this invention are derived from biological systems and organisms that achieve resilience and adaptability through redundant functional components, planning the use of these components to achieve a given task, and learning appropriateness of components from experience.")* to track and reward said one function that is selected to determine an approximate optimal operation of said computer program based on said selection command (col.8 lines 49-57 *"An enabled action selection rule is used to select one of several enabled actions for execution. Different selection rules can be used, including a trivial selection rule that picks one of the enabled actions at random. This invention specifies an enabled action preference policy that is described below with respect to FIG.* 13. ")." (3/6/2009 Communication, p. 5, ll. 2-

10) The 3/6/2009 Communication states that "[b]ut Polak does not disclose • setting at least one feedback point for each point of choice and each function from said list of pre-selected functions is associated with a reward." (3/6/2009 Communication, p. 6, ll. 10-12)

11) The 3/6/2009 Communication states that "[h]owever, Li discloses • setting at least one feedback point for each point of choice (col. 4 lines 46-52 *"An additional object of the invention is to provide self-optimizing machine and method which actively computes, and automatically sets at, the instantaneous optimal combinations of the many relevant variables in various categories, with instant feed-back to supply data for immediate replanning, retesting, and reoptimiizng")."* (3/6/2009 Communication, p. 6, ll. 13-18)

12) The 3/6/2009 Communication states that "[p]er claim 8 the rejection of claim 7 is incorporated and Polak further discloses comprising allowing a learning protocol in said computer program to determine an approximate optimal operation

of said computer program based on said specifying (col.8 lines 11-12 *"This invention preferably uses a specific type of selection rule, called a preference policy",* defining (col.5 lines 59-64 " *... The actions 20 define the functional components of the dependency action system in terms of the data that is required and the data that is produced.",* and setting (col. 13 lines 13-15 *"The dependency action system control routine begins in step 81000. Control then continues to steps 1100, where the dependency action system is initialized by setting all storage locations to "empty".")"* (3/6/2009 Communication, p. 7, ll. 5-14)

13) The 3/6/2009 Communication states that "[p]er claim 11 the rejection of claim 8 is incorporated and Polak further discloses set of alternate choices include operational choices, further comprising: inputting a rule command into said computer program, wherein said rule command establishes a rule on how to determine said approximate optimal operation (col.8 lines *11-15 "This invention preferably uses a specific type of selection rule, called a preference policy, that is based on a preference relation that specifies ifone action is to be preferred over another action").*" (3/6/2009 Communication, p. 7, ll. 15 – p. 8. ll. 2)

14) The 3/6/2009 Communication states that "[p]er claim 12 the rejection of claim 8 is incorporated and Li further discloses set of alternate choices include operational choices, and wherein said method further comprises: • inputting a reward command into said computer program at a point of choice, determined by a programmer, wherein said reward command provides reward in said computer program, which results in said approximate optimal choice for optimizing said computer program (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle.* ")." (3/6/2009 Communication, p. 8, ll. 3-11)

15) The 3/6/2009 Communication states that "[p]er claim 23 the rejection of claim 19 is incorporated Li further discloses • input a rule command into said computer program that runs on said computer, said rule command established a rule for said computer program on how to determine said approximate optimal choice of operation (col. 16 lines 55-57 *"self-optimizing machine readily and automatically generates these and other similar rules in computer-coded form ready for use as instant machine expert systems ...* ")." (3/6/2009 Communication, p. 10, ll. 15-21)

16) The 3/6/2009 Communication states that "[p]er claim 24 the rejection of claim 19 is incorporated Li further discloses • input a reward command that provides a reward, at a point of choice in said computer program which runs on said computer, for said one function selected by said selection command, which results in said approximate optimal choice for self- optimizing said computer program (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle.* ")." (3/6/2009 Communication, p. 11, ll. 1-8)

IX. ARGUMENT

A. Rejection of claims 1, 5-8, 11-13, and 17-18 as being unpatentable under 35 U.S.C. §103(a) over Polak, in view of Li.

1. Appellant's arguments with respect to Independent Claims 1, 7, and 13.

The 3/6/2009 Communication states that "[i]nputting a selection command that selects one function from a list of pre- selected functions for input into said computer program at a point of choice determined by a programmer (Co1.7 lines 54-59 *"If multiple actions are enabled, the dependency action system uses a selection rule to determine which of the enabled actions is to be executed. It is possible to use a selection rule based on random choice. A policy can be pre-established that prescribes which action of a set of enabled actions is to be selected for execution.* " & col. 12 line 66 "*The enabled action selection rule determines which of the enabled actions 620 and* 622 *is tried first. Similarly, if the action 620 does not update the storage location* 632, *but updates the storage location 630, the selection rule determines whether the action 640 or the action* 622 *is used for recovery. This behavior can be change by the designer of the dependency action system by adding additional dependencies.* "). (Fact #1)

Appellants respectfully submit that Polak fails to disclose, teach or even suggest at least the features of "...inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program at a point of choice determined by a programmer, wherein each function from said list of pre-selected functions is associated with a reward..." as recited in

independent claim 1 and similarly recited in independent claim 13. In particular, the asserted mapping fails to provide "a point of choice determined by a programmer". As note, Polak at most describes an ordering which is simply not the same as a point of choice determined by a programmer.

The 3/6/2009 Communication states that "[a]llowing a learning protocol comprised of learning instructions in said computer program (col. 4 lines 45-50 *"The method and system of this invention are derived from biological systems and organisms that achieve resilience and adaptability through redundant functional components, planning the use of these components to achieve a given task, and learning appropriateness of components from experience.")* to track and reward said one function that is selected and to determine an approximate optimal policy (sic) of choice of operation of said computer program based on said selection command (col.8 lines 49-57 *"An enabled action selection rule is used to select one of several enabled actions for execution. Different selection rules can be used, including a trivial selection rule that picks one of the enabled actions at random. This invention specifies an enabled action preference policy that is described below with respect to FIG.* 13. ")." (Fact #2)

Appellants acknowledge the resilience of biological systems and organisms

but respectfully traverse the relevance in disclosing, teaching or even suggesting "[a]llowing a learning protocol comprised of learning instructions in said computer program", as recited in independent claim 1 and similarly recited in claim 13. Moreover, the Communication fails to address at least the features of the claims which recite "...to track and reward said one function that is selected and to determine an approximate optimal choice of operation of said computer program based on said selection command..."

That is, the one function is selected from a list of pre-selected functions ... at a point chosen by a programmer..." Appellants submit there is simply no indication that Polak provides such a point of choice. Furthermore, neither the asserted biological systems and organisms nor the Polak system discloses a learning protocol comprised of learning instructions in said computer program.

The 3/6/2009 Communication states that "[b]ut Polak does not disclose • each function from said list of pre- selected functions is associated with a reward." (Fact #3) Appellants agree.

The 3/6/2009 Communication states that "[h]owever, Li explicitly discloses • wherein each function from said list of pre- selected functions is associated with a reward (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple*

reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle. ")." (Fact #4)

Appellants note that the reward recited in independent claims is associated with a function which is then used ("...track and reward...") with the learning protocol. In contrast, Li merely describes a simplification of choices that are presented to the user or manager "[t]o select to decide, e.g., whether to continue the next self-optimizing cycle." (Li, col. 16, ll. 5-9) Thus, independent claims 1, 7 and 13 define patentable subject matter over the art of record.

The 3/6/2009 Communication states that "[p]er claim 7 Polak discloses A method of optimizing a computer program, said method comprising: • specifying at least one point of choice, determined by a programmer, in said computer program (col.8 lines 11-15 *"This invention preferably uses a specific type of selection rule, called a preference policy, that is based on a preference relation that specifies if one action is to be preferred over another action")."* (Fact #7)

However, once again Appellants draw attention to the lack of the requisite "...point of choice, determined by a programmer..." in Li.

The 3/6/2009 Communication states "[d]efining a set of alternate choices at each point of choice ([Abstract] *"if one action fails to produce a value to a storage*

location, other alternative actions may still be enabled and can be executed. "), wherein said set of alternate choices include operational choices comprises:• inputting a selection command that selects one function from a list of pre-selected functions into said computer program (col.7 lines 54-57 *"If multiple actions are enabled, the dependency action system uses a selection rule to determine which of the enabled actions is to be executed. It is possible to use a selection rule based on random choice.")"* (Fact #8)

However, Appellants note that since Polak discloses no point of choice, it cannot disclose "...defining a set of alternate choices at each point of choice, wherein..." Moreover, Polak discloses no "...inputting a selection command..." as recited in the claims. Instead, Polak at most selects. Thus, claim 7 defines patentable subject matter over the art of record.

The 3/6/2009 Communication states that "[a]llowing a learning protocol comprised of learning instructions in said computer program (col. 4 lines 45-50 *"The method and system of this invention are derived from biological systems and organisms that achieve resilience and adaptability through redundant functional components, planning the use of these components to achieve a given task, and learning appropriateness of components from experience.")* to track and reward

said one function that is selected to determine an approximate optimal operation of said computer program based on said selection command (col.8 lines 49-57 *"An enabled action selection rule is used to select one of several enabled actions for execution. Different selection rules can be used, including a trivial selection rule that picks one of the enabled actions at random. This invention specifies an enabled action preference policy that is described below with respect to FIG.* 13. ")." (Fact #9)

However, Appellants note that Fig. 13; S1440 and S1450 clearly select some action for which the smallest number N was determined. Thus, Polak fails to disclose, teach or even suggest "...allowing a learning protocol..." as recited in claim 7. Instead, Li merely describes the selection of minimum steps N.

The 3/6/2009 Communication states that "[b]ut Polak does not disclose • setting at least one feedback point for each point of choice and each function from said list of pre-selected functions is associated with a reward." (Fact #10). Appellants agree.

The 3/6/2009 Communication states that "[h]owever, Li discloses • setting at least one feedback point for each point of choice (col. 4 lines 46-52 *"An additional object of the invention is to provide self-optimizing machine and method which*

actively computes, and automatically sets at, the instantaneous optimal combinations of the many relevant variables in various categories, with instant feed-back to supply data for immediate replanning, retesting, and reoptimiizng")."

(Fact #11)

Appellants respectfully submit that the cited portion of Li fails to provide any indication of the point of choice recited in the claims. Instead, the omitted but relevant last clause in the quoted sentence specifically modifies the sentence to state that "...all without human intervention..." (Li, col. 4, ll. 52) Appellants provide the omitted modifier in context for the convenience of the Board as follows:

An additional object of the invention is to provide selfoptimizing machine and method which actively computes, and automatically sets at, the instantaneous optimal combinations of the many relevant variables in various categories, with instant feed-back to supply data for immediate replanning, retesting, and reoptimiizng, all without human intervention;" (Li, col. 4. ll. 46-52, emphasis added)

Thus, it will be apparent that in fact, Li teaches away from the claimed invention.

2. Appellant's arguments with respect to Dependent Claims 5-6, 8, 11-12, and 17-18.

The 3/6/2009 Communication states that "[p]er claims 5, 17 the rejection of claims 1 and 13 are incorporated respectively and Li further discloses • comprising inputting a rule command that establishes a rule for said computer program on how to determine said approximate optimal choice of operation (col. 16 lines 55-57 *"self-optimizing machine readily and automatically generates these and other similar rules in computer-coded form ready for use as instant machine expert systems* ... ")." (Fact #5)

Appellants respectfully submit that Li's "readily and automatically" generation is simply not the same as "...inputting..." as recited in the claims. Appellants note that the cited portion of Li itself indicates that no input is required. Thus, Li describes ...commercially valuable conclusions ...and other knowledge bases, derived from ... exactl nut automatically executed R&D." (Li, col. 16, ll. 14-17)

The 3/6/2009 Communication states that "[p]er claims 6, 18 the rejection of claims 1 and 13 are incorporated and Li further discloses • comprising inputting a reward command that provides a reward, at a point of choice, determined by a

programmer, in said computer program, for said one function selected by said selection command, which results in said approximate optimal choice for optimizing said computer program (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle. ")."* (Fact #6)

Appellants respectfully draw attention to the fact that Li merely discloses a system in which the computer provides multiple choices. There is again, no indication of the "...point of choice, determined by a programmer..." as recited in the claims.

The 3/6/2009 Communication states that "[p]er claim 8 the rejection of claim 7 is incorporated and Polak further discloses comprising allowing a learning protocol in said computer program to determine an approximate optimal operation of said computer program based on said specifying (col.8 lines 11-12 *"This invention preferably uses a specific type of selection rule, called a preference policy",* defining (col.5 lines 59-64 " *... The actions 20 define the functional components of the dependency action system in terms of the data that is required and the data that is produced.",* and setting (col. 13 lines 13-15 *"The dependency*

action system control routine begins in step 81000. Control then continues to steps 1100, where the dependency action system is initialized by setting all storage locations to "empty".")" (3/6/2009 Communication, p. 7, ll. 5-14)

Appellants note the continued omission of "...a point of choice..." as referenced in the specifying, defining and setting portions of the parent claims to which claim 8 refers. Thus, claim 8 defines patentable subject matter.

Moreover, the claims recited "allowing a learning protocol in said computer program..." In Polak, the preference policy is external to the dependency action and therefore fails to address at least this recited feature.

In regard claim 11, Appellants submit that neither the selection rule nor the preference policy address "...determining said optimal operation..." as recited in claim 11.

The 3/6/2009 Communication states that "[p]er claim 12 the rejection of claim 8 is incorporated and Li further discloses set of alternate choices include operational choices, and wherein said method further comprises: • inputting a reward command into said computer program at a point of choice, determined by a programmer, wherein said reward command provides reward in said computer program, which results in said approximate optimal choice for optimizing said

computer program (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle.* ")." (Fact #14)

Appellants' above-identified arguments are equally relevant here. Namely, Li fails to disclose a "point of choice" and in fact specifically states that it is directed to performing calculations without human intervention.

Instead, the portion of Li cited above actually relates to the output namely that "[t]he manager cannot analyze, even read, the very massive, complex raw experimental data, or even the summarized, computer-generated conclusions, especially in real time. But the computer can also provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide..." (Li, col. 16, ll. 2-9) Thus, claim 12 defines patentable subject matter over the art of record.

B. Rejection of Claims 19 and 23-24 as being unpatentable under 35 U.S.C. §103(a) over Polak, in view of Li, in further view of Christensen.

3. Appellant's arguments with respect to Independent Claim 19.

Appellants note that the above-identified arguments regarding the Polak-Li combination are similarly relevant here. The Communication does not assert that Christensen remedies any of the above-identified deficiencies of Polak-Li, nor does it. Thus, independent claim 19 defines patentable subject matter over Polak-Li and/or Christensen.

4. Appellant's arguments with respect to Dependent Claims 23-24.

The 3/6/2009 Communication states that "[p]er claim 23 the rejection of claim 19 is incorporated Li further discloses • input a rule command into said computer program that runs on said computer, said rule command established a rule for said computer program on how to determine said approximate optimal choice of operation (col. 16 lines 55-57 *"self-optimizing machine readily and automatically generates these and other similar rules in computer-coded form ready for use as instant machine expert systems ...* ")." (Fact #15)

Appellants respectfully submit that "these and other similar rules" simply fails to disclose, teach or even suggest at least the features of "...wherein said pre-compiler further inputs a rule command into said computer program that runs on said computer, said rule command establishing a rule for said computer program on how to determine said approximate optimal choice of operation..." In particular, Li fails to suggest this use of a pre-compiler.

The 3/6/2009 Communication states that "[p]er claim 24 the rejection of claim 19 is incorporated Li further discloses • input a reward command that provides a reward, at a point of choice in said computer program which runs on said computer, for said one function selected by said selection command, which results in said approximate optimal choice for self- optimizing said computer program (col. 16 lines 5-9 *"the computer can also provide multiple choices of simple reward-risk choices from the highly efficient experiments for the manager to select to decide, e.g., whether to continue the next self-optimizing cycle.* ")." (Fact #16)

Appellants note the arguments regarding claims 6 and 18 are similarly relevant to claim 24. The Communication does not assert that Christensen remedies any of the above-identified deficiencies of Polak and Li, nor does it. Thus, claim 24 defines patentable subject matter over Polak, Li and/or Christensen for at least the

same reasons.

X. CONCLUSION

In view of the foregoing, Appellant submits that claims 1, 5-8, 11-13, 17-19, and 23-24, all of the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance. Thus, the Board is respectfully requested to remove the rejections of claims 1, 5-8, 11-13, 17-19, and 23-24.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Attorney's Deposit Account No. 09-0441.

Respectfully submitted,

Date: August 6, 2009

/Christian Austin-Hollands/
Christian Austin-Hollands, Esq.
Registration No. 46,527

Gibb I.P. Law Firm LLC, LLC
2568-A Riva Road, Suite 304
Annapolis, MD, 21401
Voice: (410) 573-1545
Fax: (301) 261-8825
Customer No. 29154

APPENDIX

CLAIMS SECTION

1. (REJECTED) A method of instructing a computer program to self-optimize, said method comprising:

inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program at a point of choice determined by a programmer, wherein each function from said list of pre-selected functions is associated with a reward; and

allowing a learning protocol comprised of learning instructions in said computer program to track and reward said one function that is selected and to determine an approximate optimal choice of operation of said computer program based on said selection command.

2-4. (Canceled).

5. (REJECTED) The method of claim 1, further comprising inputting a rule command that establishes a rule for said computer program on how to determine said approximate optimal choice of operation.

6. (REJECTED) The method of claim 1, further comprising inputting a reward

command that provides a reward, at a point of choice, determined by a programmer, in said computer program, for said one function selected by said selection command, which results in said approximate optimal choice for self-optimizing said computer program.

7. (REJECTED) A method of optimizing a computer program, said method comprising:

specifying at least one point of choice, determined by a programmer, in said computer program;

defining a set of alternate choices at each point of choice, wherein said set of alternate choices include operational choices comprising:

inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program, wherein each function from said list of pre-selected functions is associated with a reward;

allowing a learning protocol comprised of learning instructions in said computer program to track and reward said one function that is selected to determine an approximate optimal operation of said computer program based on said selection command; and

setting at least one feedback point for said each point of choice.

8. (REJECTED) The method of claim 7, further comprising allowing a learning protocol in said computer program to determine an approximate optimal operation of said computer program based on said specifying, defining, and setting.

9-10. (Canceled).

11. (REJECTED) The method of claim 8, wherein said set of alternate choices include operational choices, further comprising:

inputting a rule command into said computer program, wherein said rule command-establishes a rule on how to determine said approximate optimal operation.

12. (REJECTED) The method of claim 8, wherein said set of alternate choices include operational choices, and wherein said method further comprises:

inputting a reward command into said computer program at a point of choice, determined by a programmer, wherein said reward command provides reward in said computer program, which results in said approximate optimal choice for optimizing said computer program.

13. (REJECTED) A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method of instructing a computer program to self-optimize, said method comprising:

inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program at a point of choice, determined by a programmer, wherein each function from said list of pre-selected functions is associated with a reward; and

allowing a learning protocol comprised of programmer determined learning

instructions in said computer program to track and reward said one function that is selected and to determine an approximate optimal choice of operation of said computer program based on at least said selection command.

14-16. (Canceled).

17. (REJECTED) The program storage device of claim 13, further comprising inputting a rule command that establishes a rule for said computer program on how to determine said approximate optimal choice of operation.

18. (REJECTED) The program storage device of claim 13, further comprising inputting a reward command that provides a reward, at a point of choice, determined by a programmer, in said computer program, for said one function selected by said selection command, which results in said approximate optimal choice for optimizing said computer program.

19. (REJECTED) A computer system comprising:

a pre-compiler that inputs a selection command at a point of choice, determined by a programmer, into a computer program that runs on a computer, said selection command selecting one function from a list of pre-selected functions for input into said computer program, wherein each function from said list of pre-selected functions is associated with a reward; and

a processor adapted to execute a learning protocol in said computer program to track and reward said one function that is selected and determine an approximate

optimal operation of said computer program based on at least said selection command.

20-22. (Canceled).

23. (REJECTED) The system of claim 19, wherein said pre-compiler further inputs a rule command into said computer program that runs on said computer, said rule command establishing a rule for said computer program on how to determine said approximate optimal choice of operation.

24. (REJECTED) The system of claim 19, wherein said pre-compiler further inputs a reward command that provides a reward, at a point of choice in said computer program which runs on said computer, for said one function selected by said selection command, which results in said approximate optimal choice for optimizing said computer program.

25. (Canceled).

APPENDIX

CLAIM SUPPORT AND DRAWING ANALYSIS SECTION

The following annotated claims help illustrate the features defined by the claims, but are not intended to be an exhaustive listing of the claimed features. Instead, the specification may contain many more examples of such claimed features. Thus, the following claims are not intended to be limited to or limited by the following brief annotations.

1. A method of instructing a computer program to self-optimize, said method comprising: **(1)** inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program at a point of choice determined by a programmer, wherein each function from said list of pre-selected functions is associated with a reward **{Fig. 3, element 350; paragraph [0019]}**; and **(2)** allowing a learning protocol comprised of learning instructions in said computer program to track and reward said one function that is selected and to determine an approximate optimal choice of operation of said computer program based on said selection command **{Fig. 3, element 340 paragraph [0019]}**.

7. A method of optimizing a computer program, said method comprising: **(1)** specifying at least one point of choice, determined by a programmer, in said

computer program **{Fig. 3, element 310, paragraph [0019]}**; (**2**) defining a set of alternate choices at each point of choice **{Fig. 3, element 320 paragraph [0019]}**, wherein said set of alternate choices include operational choices comprising: (**3**) inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program, wherein each function from said list of pre-selected functions is associated with a reward **{Fig. 3, element 350 paragraph [0019]}**; (**4**) allowing a learning protocol comprised of learning instructions in said computer program to track and reward said one function that is selected to determine an approximate optimal operation of said computer program based on said selection command **{Fig. 3, element 340 paragraph [0019]}**; and (**5**) setting at least one feedback point for said each point of choice **{Fig. 3, element 330 paragraph [0019]}**.

13. A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method of instructing a computer program to self-optimize, said method comprising: (**1**) inputting a selection command that selects one function from a list of pre-selected functions for input into said computer program at a point of choice, determined by a programmer, wherein each function from said list of pre-selected functions is associated with a reward **{Fig. 3, element 350 paragraph [0019]}**; and (**2**) allowing a learning protocol comprised of programmer determined learning instructions in said computer program to track and reward said one function that is selected and to determine an approximate optimal choice of operation of said computer program based on at least said selection command **{Fig. 3, element 340**

paragraph [0019]}.

19. A computer system comprising: **(1)** a pre-compiler that inputs a selection command at a point of choice, determined by a programmer, into a computer program that runs on a computer, said selection command selecting one function from a list of pre-selected functions for input into said computer program, wherein each function from said list of pre-selected functions is associated with a reward **{Fig. 3, element 350 paragraph [0019]}**; and **(2)** a processor adapted to execute a learning protocol in said computer program to track and reward said one function that is selected and determine an approximate optimal operation of said computer program based on at least said selection command **{Fig. 3, element 340 paragraph [0019]}**.

APPENDIX

MEANS OR STEP PLUS FUNCTION ANALYSIS SECTION

Not applicable.

APPENDIX

EVIDENCE SECTION

Not applicable.

APPENDIX

RELATED CASES SECTION

Not applicable.